What is claimed is:

1. A computer system with improved tolerance to microprocessor functional interrupts induced by environmental sources, comprising:

a microprocessor;

an array of memory, volatile or non-volatile, connected to said microprocessor;

a hardened core circuit, designed to withstand environmentally induced faults, and connected to said microprocessor, in a manner allowing for said microprocessor's interrupt control, reset control, data bus, and address bus signals to connect to said hardened core circuit, and for said hardened core's status, interrupt output and power cycle output signals to connect to said microprocessor;

a microprocessor software routine configured to send a predetermined timer signal from the microprocessor to the said hardened core circuit on a predetermined time period;

a hardened core circuit function configured to read predetermined timer signal from said microprocessor on predetermined time period and activate said microprocessor's interrupt and reset control input signals if timer signal is not received within predetermined time period to provide for removal of said microprocessor from functionally interrupted state;

a microprocessor software routine located at said microprocessor's interrupt or reset vector addresses, configured to restart the microprocessor's application software.

2. A system of claim 1 further comprising a microprocessor software routine configured to send maintenance data to microprocessor memory prior to functional interrupt and configured to read said maintenance data from microprocessor memory after microprocessor's removal from functionally interrupted state and use maintenance data to restart microprocessor's application software routines.

3. The system of claim 2 further comprising a microprocessor software routine configured to read said hardened core status signal(s), and to determine if interrupt or reset activation was result of hardened core activation and then restart application software routines, or normal interrupt or reset and then continue with normal application software operation.

4. The system of claim 3 further comprising a microprocessor software routine configured to to halt all currently operating application software threads.

5. The system of claim 4 further comprising a microprocessor software routine configured to read hardened core status signal(s), and to determine if multiple functional interrupts occurred within predetermined time period and then to restart all microprocessor software and hardware if multiple functional interrupts occurred within predetermined time period, or, if single functional interrupt occurred in predetermined time period to then to read maintenance data stored in said memory and provide a controlled restart of selected application software.

6. A computer system with improved fault tolerance from microprocessor data errors and functional interrupts, comprising:

a microprocessor;

an array of memory, volatile or non-volatile, connected to said microprocessor;

a fault tolerant software routine configured to send a first instruction and at least a second instruction to the microprocessor, the first and at least the second instructions being identical and spatially separated functional computational units of the VLIW microprocessor in at different clock cycles;

a first and at least a second memory device in communication with the microprocessor, the first memory device configured to store the first instruction, the second memory device configured to store at least the second instruction;

a software instruction to compare the first instruction to at least the second instruction;

a comparator to compare the first instruction to the second instruction;

a hardened core circuit, designed to withstand environmentally induced faults, and connected to said microprocessor, in a manner allowing for said microprocessor's interrupt control, reset control, data bus, and address bus signals to connect to said hardened core circuit, and for said hardened core's status, interrupt output and power cycle output signals to connect to said microprocessor;

a microprocessor software routine configured to send a predetermined timer signal from the microprocessor to the said hardened core circuit on a predetermined time period;

a hardened core circuit function configured to read predetermined timer signal from said microprocessor on predetermined time period and activate said microprocessor's interrupt and reset control input signals if timer signal is not received within predetermined time period to provide for removal of said microprocessor from functionally interrupted state; and

a microprocessor software routine located at said microprocessor's interrupt or reset vector addresses, configured to restart the microprocessor's application software.

7. The system of claim 6 further comprising a third instruction sent by the fault tolerant software routine to the microprocessor, the third instruction stored in a third memory device in communication with the microprocessor.

8. The system of claim 7 wherein the software instruction directs the comparator to compare the first, second, and third instruction.

9. The system of claim 8 wherein a match of the any of the first, second, and third instructions is accepted by the microprocessor.

10. The system of claim 6 wherein the microprocessor comprises a VLIW microprocessor.

11. A software and hardware computer system with improved fault tolerance from microprocessor data errors and functional interrupts, comprising:

a very long instruction word microprocessor;

an array of memory, volatile or non-volatile, connected to said microprocessor;

a fault tolerant software routine comprising a first instruction and a second instruction, each inserted into two spatially separate functional computational units in the VLIW microprocessor at two different clock cycles and stored in a memory device in communication with the microprocessor, the first and second instructions being identical;

a software instruction to compare the first and second instruction in the memory device in communication with a VLIW microprocessor compare or branch units, and configured to perform an action if the first and second instruction match,

the fault tolerant software routine comprising a third inserted into a third spatially separate functional computational units in the VLIW microprocessor at a third different clock cycles and stored in a third memory device in communication with the microprocessor, the first, second, and third instructions being identical;

the software instruction to compare the first, second, and third instructions in the memory devices in communication with a VLIW microprocessor compare or branch units, and configured to perform an action if any of the first, second and third instructions match;

a hardened core circuit, designed to withstand environmentally induced faults, and connected to said microprocessor, in a manner allowing for said microprocessor's interrupt control, reset control, data bus, and address bus signals to connect to said hardened core circuit, and for said hardened core's status, interrupt output and power cycle output signals to connect to said microprocessor;

a microprocessor software routine configured to send a predetermined timer signal from the microprocessor to the said hardened core circuit on a predetermined time period;

a hardened core circuit function configured to read predetermined timer signal from said microprocessor on predetermined time period and activate said microprocessor's interrupt and reset control input signals if timer signal is not received within predetermined time period to provide for removal of said microprocessor from functionally interrupted state; and

a microprocessor software routine located at said microprocessor's interrupt or reset vector addresses, configured to restart the microprocessor's application software.

12. A method of processing data in a fault tolerant computer system, comprising:

generating a first instruction at a first time interval;

generating a second instruction identical to the first instruction at a second time interval;

generating a third instruction identical to the first and second instructions at a third time interval;

comparing the first, second and third instructions;

matching anyone of the first, second, or third instructions to each other; and

performing an action based on the match instruction.

13. A method of processing data in a fault tolerant computer system, comprising:

generating a first instruction at a first time interval;

generating a second instruction identical to the first instruction at a second time interval;

comparing the first and second instructions to each other;

performing an action based on the matched first and second instructions;

generating a third instruction identical to the first and second instructions at a third time interval is the first and second instructions do not match;

matching the first, second, and third instructions to each other; and

performing an action based on a match between anyone of the first, second, and third instructions.